



INFOMATEK

Volume 1 Nomor 2 September 1999

JURNAL INFORMATIKA, MANAJEMEN DAN TEKNOLOGI

ROUTING PADA MULTIKOMPUTER BERTOPOLOGI
HYPERCUBE

Dwi Nurchayati

PENDEKATAN HEURISTIK UNTUK PENJADWALAN PRODUK-
PRODUK BERSTRUKTUR MULTI-LEVEL

Tjutju T. Dimiyati

ALLOY DESIGN AND CASTING PRACTICE OF ABRASION
AUSTENITIC MANGANESE STEEL

Muki Satya Permana

KAJI TEORITIK MESIN REFRIJERASI UNTUK INSTALASI
PEMBUAT ES BALOK (ICE-CUBE PLANT) KAPASITAS 4 TON

Toto Supriyono, Sugiharto, Hery Sonawan

PERANAN BIOTEKNOLOGI INDUSTRI DALAM UPAYA
PENGEMBANGAN KOMODITAS HASIL PERTANIAN

Wisnu Cahyadi

FAKULTAS TEKNIK UNIVERSITAS PASUNDAN BANDUNG



DAFTAR ISI

Dwi Nurchayati 41 - 55 ROUTING PADA MULTIKOMPUTER BERTOPOLOGI HYPERCUBE

Tjutju T. Dimiyati 57 - 66 PENDEKATAN HEURISTIK UNTUK PENJADWALAN PRODUK-PRODUK BERSTRUKTUR MULTI-LEVEL

Muki Satya Permana 67 - 76 ALLOY DESIGN AND CASTING PRACTICE OF ABRASION-AUSTENITIC MANGANESE STEEL

Toto Supriyono, Sugiharto, Hery Sonawan 77 - 83 KAJI TEORITIK MESIN REFRIJERASI UNTUK INSTALASI PEMBUAT ES BALOK (ICE-CUBE PLANT) KAPASITAS 4 TON

Wisnu Cahyadi 85 - 90 PERANAN BIOTEKNOLOGI INDUSTRI DALAM UPAYA PENGEMBANGAN KOMODITAS HASIL PERTANIAN



INFOMATEK

Volume 1 Nomor 2 September 1999

PENDEKATAN HEURISTIK UNTUK PENJADWALAN PRODUK-PRODUK BERSTRUKTUR MULTI-LEVEL

Tjutju T. Dimiyati*

Laboratorium Sistem Produksi Jurusan Teknik dan Manajemen Industri
Fakultas Teknik – Universitas Pasundan Bandung

Abstrak

Makalah ini membahas suatu pendekatan heuristik untuk menyelesaikan pengaturan operasi-operasi jobshop dalam pembuatan sejumlah produk yang berstruktur multi-level. Pendekatan heuristik yang dikembangkan dimaksudkan untuk menghilangkan *disjunctive constraint*, yaitu pembatas yang menyatakan hubungan di antara operasi-operasi dari produk/komponen yang berbeda yang diproses pada mesin yang sama, dan menggantinya dengan hubungan kebergantungan antar operasi (*precedence constraints*). Dengan menggunakan kriteria *minimum makespan*, pendekatan yang dikembangkan digunakan untuk menyelesaikan sejumlah contoh numerik. Hasil perhitungan disajikan untuk menggambarkan performansi dan metoda pendekatan ini.

Kata kunci : proses jobshop, *disjunctive constraint*, *precedence constraint*, pendekatan heuristik.

1. PENDAHULUAN

Sejak awal tahun 1950-an model dan metoda penyelesaian untuk persoalan penjadwalan telah menjadi *research interest*, sehingga mengalami perkembangan pesat. Hal ini antara lain disebabkan karena persoalan ini berada pada kelompok persoalan *NP-Hard*, dimana jumlah urutan yang dapat terjadi akan

bertambah secara eksponensial jika ukuran persoalannya bertambah besar. Penelitian menunjukkan bahwa untuk persoalan penjadwalan jobshop, hanya ada dua kasus yang waktu penyelesaiannya bersifat *polynomial*, yaitu kasus khusus dimana jumlah mesin dan jumlah pekerjaan lebih kecil atau sama dengan dua. Lenstra et al. [1] dan Fisher et al [2]

Karena solusi optimal untuk persoalan penjadwalan hanya mungkin diperoleh untuk persoalan yang berukuran kecil dan melibatkan ongkos serta prosedur enumerasi yang tidak praktis, maka model dan metoda penyelesaian harus terus dikembangkan, supaya persoalan-persoalan nyata yang pada umumnya berukuran besar dapat diselesaikan. Salah satu metoda pendekatan yang banyak dikembangkan adalah metoda yang didasarkan pada aturan-aturan heuristik. Metoda pendekatan ini menyebabkan persoalan dapat diselesaikan dengan cara yang lebih sederhana, tetapi tidak ada jaminan bahwa solusinya optimal.

Makalah ini membahas pengembangan suatu metoda pendekatan heuristik untuk menyelesaikan persoalan pengurutan operasi-operasi dari suatu proses jobshop, dalam pembuatan sejumlah produk yang berstruktur multi-level. Aturan heuristik yang digunakan dimaksudkan untuk menentukan prioritas pengerjaan bagi operasi-operasi yang berada pada level yang sama dan bersifat *disjunctive*, sedang kriteria penjadwalannya adalah minimum waktu penyelesaian seluruh pekerjaan (*minimum makespan*).

II. PENGEMBANGAN MODEL

2.1 Notasi yang digunakan

N Set dari produk akhir yang akan dijadwalkan. ($i = 1, \dots, N$)

M Set dari mesin yang terlibat
 P_{i0} menyatakan produk akhir i yang akan dijadwalkan ; $i = 1, \dots, N$
 P_{ij} menyatakan komponen ke j dari produk akhir i
 $Z(P_{ij})$ set dari induk-induk komponen j hingga produk akhir i
 H_{ij} banyaknya komponen j yang dibutuhkan untuk membuat satu unit induk langsungnya
 B_i Jumlah unit produk i yang harus dijadwalkan
 Q_{ij} kuantitas P_{ij} yang harus dibuat untuk memenuhi kebutuhan
 $Q_{ij} = B_i * H_{ij} (\prod H_{ik})$
 $P_{ik} \in Z(P_{ij})$

dengan $H_{ik} \geq 1$
 $P_{ik} \in Z(P_{ij})$

O_{ijk} operasi ke- k untuk P_{ij} yang dilakukan di mesin m ; $m = 1, \dots, M$

t_{ijk} waktu dimulainya operasi O_{ijk}

o_{ijk} waktu pemrosesan per unit part yang diperlukan oleh operasi O_{ijk}

h_{ijk} harga operasi O_{ijk} yang dibuat
 Mengandung nilai besar.

u_{ijk} biaya pemrosesan operasi O_{ijk} yang dibuat
 Mengandung nilai besar
 Mengandung nilai bernilai 1, jika $h_{ijk} > 0$ atau bernilai 0, jika sebaliknya

su_m waktu set-up mesin m

R_m waktu mesin m untuk digunakan
 Mengandung nilai bernilai 1, jika mesin m digunakan untuk operasi pertama setiap produk

2.2 Pengembangan Model

Model matematis untuk persoalan ini telah dibangun dan diuji coba dengan data nyata dan telah

diformulasikan sebagai model **Mixed Integer Linear Programming (MILP)**.

Formulasi tersebut, Dimiyati [3], adalah :

$$F1: \text{Min } F \quad (1)$$

dengan pembatas

$$S_{ijm} \geq R_{im}, \forall m; \quad (2)$$

$$C_{ijk} - S_{ijk} \leq 0, \forall i, j, k, m; \quad (3)$$

$$C_{ijk} - S_{ijk} \leq 0, \forall i, j, k, m; \quad (4)$$

$$C_{ijk} - F < 0, \forall i, j, k, m; \quad (5)$$

$$C_{ijk} - S_{ijk} = Q_{ijk}, \forall i, j, k, m; \quad (6)$$

$$C_{ijk} - S_{ijk} = Q_{ijk}, \forall i, j, k, m; \quad (7)$$

$$C_{ijk} - S_{ijk} = Q_{ijk}, \forall i, j, k, m; \quad (7a)$$

$$C_{ijk} - S_{ijk} = Q_{ijk}, \forall i, j, k, m; \quad (8)$$

$$X_{ijkqm} \in \{0, 1\}, \forall i, j, k, q, r, s, m; \quad (8)$$

$$F, C_{ijk}, S_{ijk} \geq 0, \text{ integer}$$

Seperti diketahui, banyaknya variabel biner dalam suatu formulasi MILP akan mempengaruhi tingkat kesulitan dan waktu penyelesaiannya. Karena itu cara yang efisien untuk menyelesaikan persoalan ini adalah dengan menetapkan nilai variabel-variabel biner dari pembatas disjunctive sehingga menjadi hubungan kebergantungan antar operasi-operasi yang bersaing dalam menggunakan suatu mesin. Dengan cara seperti ini maka ukuran persoalan dapat diperkecil sehingga lebih mudah diselesaikan, meskipun solusinya mungkin hanya mendekati optimal.

Untuk penjadwalan jobshop dengan produk-produk yang berstruktur satu-level, cara seperti ini telah dilakukan oleh Greenberg [4]. Dengan melakukan relaksasi terhadap pembatas disjunctive yang menyebabkan munculnya variabel biner, maka persoalan dapat diselesaikan sebagai persoalan **Linear Programming**. Jika hasil yang diperoleh menunjukkan tidak terjadinya konflik dalam penggunaan mesin, maka jadwal optimal telah diperoleh. Jika terjadi konflik misalnya antara operasi j dan operasi r pada mesin k , maka dibentuk dua sub-persoalan dengan menambahkan pembatas baru $C_{jk} - C_{rk} \geq t_{rk}$ pada sub-persoalan 1 dan $C_{rk} - C_{jk} \geq t_{rk}$ pada sub-persoalan yang lain. Dengan cara ini variabel biner dapat dihilangkan.

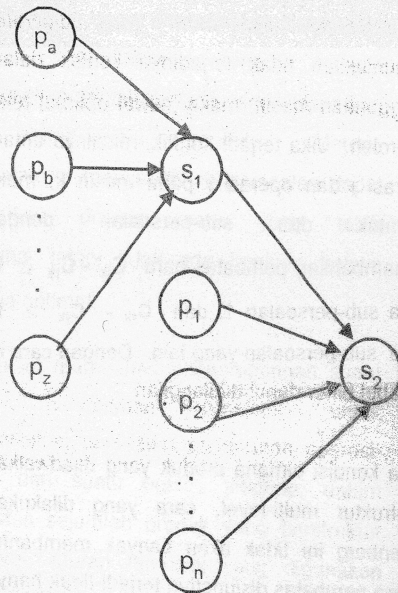
Pada kondisi dimana produk yang dijadwalkan berstruktur multi-level, cara yang dilakukan Greenberg ini tidak akan banyak membantu, karena pembatas disjunctive terjadi tidak hanya diantara operasi-operasi dari produk yang berbeda, tapi juga diantara komponen atau subassembly dalam suatu produk sehingga sub-persoalan yang terbentuk akan sangat besar jumlahnya. Karenanya, penyederhanaan persoalan dilakukan dengan menggunakan teorema berikut ini.

Teorema 1

Jika G adalah graph yang menyatakan struktur suatu produk, maka untuk menjadwalkan node-

node pada G (baik part atau subassembly) harus dimulai dari level yang paling bawah (nomor level paling besar) agar diperoleh jadwal dengan makespan minimum.

Pembuktian : Perhatikan Gambar 1.



Gambar 1. Digraph untuk Teorema 1

Berdasarkan precedence constraint antara s_1 dengan p_a, p_b, \dots, p_z dan antara s_2 dengan p_1, p_2, \dots, p_n maka penjadwalan node yang dimulai dari level paling bawah akan memberikan makespan sebagai berikut :

$$C_{\max} = \sum_{j=a}^z t(p_j) + \sum_{i=1}^n t(p_i) + t(s_2) +$$

$$\max [t(s_1) - \sum_{i=1}^n t(p_i), 0]$$

dimana $t(\bullet)$ adalah waktu proses part atau subassembly (\bullet). Jika ada sembarang $p_k, k = 1, 2, \dots, n$ dijadwalkan sebelum $p_j, j = a, b, \dots$ maka makespan adalah :

$$C_{\max} = \sum_{j=a}^z t(p_j) + \sum_{i=1}^n t(p_i) + t(s_2) + \max [t(s_1) - \sum_{i=1}^n t(p_i), 0]$$

Berdasarkan teorema di atas, maka jika p_k dan p_{qrs} adalah dua operasi yang berada pada level operasi yang berbeda (bersifat non-paralel), maka cara untuk menentukan urutannya adalah dengan mendahulukan operasi pada level yang lebih rendah. Untuk mengatasi persaingan penggunaan mesin oleh operasi-operasi yang berada pada level operasi yang sama (bersifat paralel), maka perlu diterapkan aturan prioritas tertentu, sehingga pembatas disjunctive (7) dan (7a) dapat diganti dengan pertidaksamaan yang sederhana.

Sehingga demikian, formulasi persoalan semula menjadi F2 yang pembatas (7) dan (7a) akan menjadi $C_{ijkm} - S_{qrsn} \leq 0$, jika O_{ijkm} mendahului O_{qrsn} .

III. ALGORITMA HEURISTIK

Langkah-langkah yang dilakukan untuk menyelesaikan persoalan secara heuristik adalah sebagai berikut :

data dan input data berupa produk-produk yang akan dijadwalkan, lengkap dengan data kebutuhan, struktur produk, dan urutan proses pengerjaan untuk masing-masing produk (*routing*).

2. Berdasarkan data di atas, buat *Bill of Operations*, yaitu uraian setiap produk ke dalam komponen-komponen masing-masing serta operasi-operasi yang harus dilakukan. Tentukan level operasi dan lokasi dengan cara yang paling efisien untuk menentukan *level* dan *lokasi*.

3. Kelengkapan proses operasi yang akan dilakukan, apakah dari bahan baku ke produk jadi, atau sebaliknya, atau ada yang lain. Tentukan *level* dan *lokasi* yang sesuai.

4. Karir yang akan dilakukan, apakah akan dilakukan di satu level saja, atau akan dilakukan di beberapa level. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai.

5. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai.

6. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai.

7. Pada setiap level operasi, susunlah operasi-operasi mulai dari operasi dengan R_{ijkm} terbesar (sebagai aturan, prioritas

yang dipilih). Jika ada operasi yang dilakukan di satu level saja, atau akan dilakukan di beberapa level.

8. Khusus pada level yang paling rendah, apakah ada operasi operasi yang dilakukan di level yang sama. Jika ya dan operasi tersebut dilakukan pada mesin yang sama, susunlah operasi-operasi tersebut berdasarkan waktu proses. Tentukan *level* dan *lokasi* yang sesuai.

9. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai.

10. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai.

11. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai.

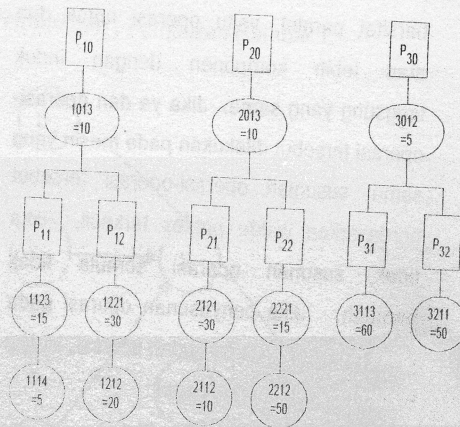
12. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai.

13. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai.

14. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai.

15. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai. Tentukan *level* dan *lokasi* yang sesuai.

Sebagai ilustrasi dari algoritma di atas, perhatikan contoh sederhana berikut ini. Misalkan Bill-of-Operations dari tiga jenis produk yang dijadwalkan adalah sebagai berikut :



Gambar 2. Bill-of-Operations

Hasil 5 langkah pertama terlihat pada Tabel 1.

Langkah 6 : Operasi O₂₂₁₂ dan O₂₁₁₂ adalah operasi yang bersifat paralel, dan karena urutannya tidak berdasarkan waktu proses terkecil maka posisinya harus ditukar

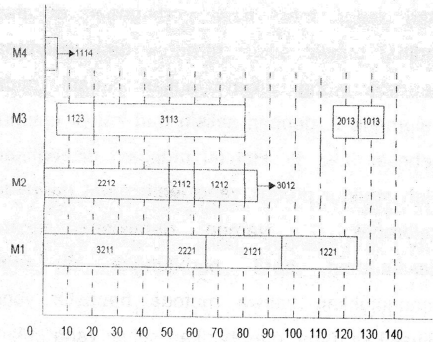
Hasil perhitungan hingga langkah 8 adalah seperti pada Lampiran 1, sehingga pada langkah 9 diperoleh pembatas baru penggan pembatas (7) dan (7a) sebagai berikut :

$$\begin{aligned}
 C_{3211} - S_{2121} &\leq 0 \\
 C_{2121} - S_{1221} &\leq 0 \\
 C_{1221} - S_{2221} &\leq 0 \\
 C_{2112} - S_{1212} &\leq 0 \\
 C_{1212} - S_{2212} &\leq 0 \\
 C_{2212} - S_{3012} &\leq 0 \\
 C_{3113} - S_{1123} &\leq 0 \\
 C_{1123} - S_{1013} &\leq 0 \\
 C_{1013} - S_{2013} &\leq 0
 \end{aligned}$$

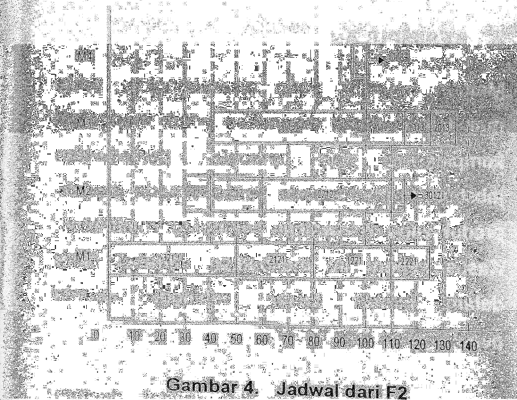
Dengan tidak menghilangkan kemungkinan lainnya, R_m dan su_m ditetapkan sama dengan 0. Formulasi F2 dapat dilihat pada Lampiran 2 yang diselesaikan dengan menggunakan software *LINDO-Hyper*. Hasil perhitungan ini memberikan solusi yang sama dengan solusi optimal dari F1, yaitu makespan sebesar 135, seperti terlihat pada Gambar 3 dan Gambar 4.

Tabel 1. Hasil perhitungan langkah 1 sampai dengan langkah 5

Level Opr.	MESIN-1			MESIN-2			MESIN-3			MESIN-4		
	O _{ijkm}	R _{tijkm}	T _{ijkm}	O _{ijkm}	R _{tijkm}	T _{ijkm}	O _{ijkm}	R _{tijkm}	T _{ijkm}	O _{ijkm}	R _{tijkm}	T _{ijkm}
2	3211	55	50	2212	75	50	3113	65	60	1114	30	5
1	1221	40	30	1212	50	20						
	2121	40	30	2112	50	10						
0	2221	25	15	3012	5	5	1123	25	15			
							1013	10	10			
							2013	10	10			



Gambar 3. Jadwal dari F1



Gambar 4. Jadwal dari F2

IV. ANALISA PERFORMANSI

Untuk mengetahui performansi dari metoda heuristik yang dikembangkan, sebanyak 22 soal telah diselesaikan. Soal-soal tersebut dibagi ke dalam tiga kelompok berdasarkan kompleksitas dari struktur produk-produk yang dijadwalkan. Kelompok 1 terdiri dari 5 soal dengan jumlah level dari struktur produk tidak lebih dari 3 untuk

masing-masing produk. Kelompok 2 terdiri dari 11 soal dengan struktur dari produk-produknya paling sedikit terdiri dari 4 level. Kelompok 3 terdiri dari 6 soal, dimana produk-produk yang dijadwalkan memiliki struktur dengan jumlah level yang merupakan campuran dari kelompok 1 dan 2.

Jumlah produk yang dijadwalkan pada masing-masing kelompok soal sedikitnya adalah 2, sedang jumlah komponen untuk setiap subassembly pada masing-masing kelompok soal adalah antara 2 hingga 4 untuk kelompok 1, dibatasi sebanyak 2 untuk kelompok 2, dan campuran untuk kelompok 3. Data jumlah operasi untuk setiap komponen/subassembly diturunkan dari distribusi Uniform dengan parameter $U(1-4)$ sedang data waktu proses setiap operasi diturunkan dari distribusi yang sama dengan parameter $U(0.01-3.0)$. Data kebutuhan komponen untuk setiap produk akhir diturunkan dari $U(10-200)$.

Seluruh persoalan diselesaikan dengan menggunakan software LINDO-Hyper dan asumsi nilai $R_m = 0$. Hasil perhitungan untuk masing-masing kelompok persoalan disajikan pada Tabel 2, Tabel 3, dan Tabel 4. Ketiga tabel memperlihatkan hasil perhitungan untuk formulasi F1 yang merupakan solusi optimal dari persoalan, dan hasil perhitungan untuk formulasi F2 yang merupakan solusi dari metoda heuristik yang dikembangkan.

Tabel 2 Hasil perhitungan untuk kelompok 1

Soal no.	Sol. Opt dari F1	Sol. Heur dari F2	Selisih (%)
1	89	90	1.12
2	91	92	1.09
3	94	95	1.06
4	98	98	0
5	103	103	0

Tabel 3 Hasil perhitungan untuk kelompok 2

Soal no.	Sol. Opt dari F1	Sol. Heur dari F2	Selisih (%)
1	101	104	2.97
2	114	117	2.63
3	102	105	2.94
4	100	101	1.00
5	106	106	0
6	107	110	2.80
7	127	132	3.93
8	121	121	0
9	127	129	1.57
10	132	135	2.27
11	134	134	0

Tabel 4 Hasil perhitungan untuk kelompok 3

Soal no.	Sol. Opt dari F1	Sol. Heur dari F2	Selisih (%)
1	120	120	0
2	205	205	0
3	158	158	0
4	136	137	0.74
5	194	197	1.54
6	166	166	0

Dari ketiga tabel hasil perhitungan di atas terlihat bahwa solusi terburuk dari algoritma heuristik yang dikembangkan terjadi pada kelompok 2, dengan selisih dari solusi optimal sebesar 3.93 %. Hal ini mungkin disebabkan oleh struktur produk yang lebih rumit daripada kelompok 1. Namun demikian, secara keseluruhan hasil perhitungan di atas menunjukkan bahwa metoda heuristik yang diusulkan dapat memberikan solusi yang cukup baik.

V. KESIMPULAN

Metoda heuristik yang dikembangkan pada makalah ini dapat digunakan untuk mengatasi kompleksitas dari persoalan penjadwalan workshop, terutama pada kasus dimana produk yang dijadwalkan berstruktur multi-level. Kasus seperti ini melibatkan variabel biner dengan jumlah besar, sehingga ukuran persoalan juga besar.

Metoda yang sederhana dan mudah dengan menggunakan algoritma heuristik ini layak untuk diimplementasikan dalam mengendalikan produksi sehari-hari di lantai produksi.

Pada penelitian ini setiap komponen atau part diasumsikan hanya mempunyai satu induk, sehingga penjadwalannya dilakukan secara individual. Kasus ini berlaku jika struktur

produk-produk yang dijadwalkan adalah multi-level dengan *type assembly*. Penelitian lebih lanjut dapat dilakukan untuk kondisi dimana produk-produk yang dijadwalkan mempunyai struktur multi-level *type general*, dimana komponen/part yang sama akan digunakan oleh beberapa produk atau subassembly yang berbeda. Pada kasus ini, part/komponen yang sama dapat dijadwalkan ke dalam satu batch.

VI. DAFTAR PUSTAKA

- [1] Lenstra, J.K., Rinnooy Kan, A.H.G., and Bruckner, P., (1977), *The Complexity of Machine Scheduling Problems*, *Annals of Discrete Mathematics*, Vol. 1, 343 - 362
- [2] Fisher, M.L., Lageweg, B.J., Lenstra, J.K., and Rinnooy Kan, A.H.G., (1983), *Surrogate Duality Relaxation for Jobshop Scheduling* *Discrete Applied Mathematics*, Vol. 5, 65-75
- [3] Dimiyati, T.T., (1999), *Model Optimasi Penjadwalan Jobshop untuk Produk-Produk Berstruktur Multi-Level*, *Infomatek*, Vol.1, 1-9
- [4] Greenberg, H.H., (1968), *A Branch and Bound Solution to the General Scheduling Problem*, *Operations Research*, Vol. 8, 353- 361